CS118
Problem Solving

How to begin a formal problem solving process?

First realize that there is a difference between The Solution and The Implementation. The Solution is what you are developing – it is a generic **plan** for solving *multiple* instances of a problem. The Implementation is when you take The Solution and use a programming language (like MATLAB or Python or Java) to make it happen. **You must prepare The Solution without assumptions about The Implementation. Do not assume anything about the language or platform in which The Implementation will occur.**

> *Analogy*: You want to build a house. So you make a set of design plans – this is The Solution. When you are ready to make the house a reality, you provide the plans to a builder who constructs the house. The builder is The Implementation.
>
> Your plans do not tell the builder that he must use hand tools, or power tools; they do not tell the builder who he must hire; they do not tell the builder where it will be built; they do not specify colors of paint. These are all "implementation details". The Solution should work for any Implementation.

Approach

You must read and understand The Problem to determine:

- the starting situation of The Solution (what are its assumptions);
- the ending situation of The Solution (what is to be accomplished);
- what constraints and information are available for The Solution.

Methods

You are providing instructions for what the The Solution is going to do. **The Solution will do nothing except what you tell it.** The Solution only has the ability to use The Tools. All other capabilities must be built from The Tools.

Data

When creating The Solution, information used by it must either be known ahead of time (e.g. the value of π or a fixed piece of information related to The Problem) or obtained via the INPUT tool.

> CAUTION: When writing The Solution remember that it must be general.
>
> For example, you could write a solution for determining the square root of a number (in fact you will later this semester). A solution which only computed the square root of a single value would be virtually useless, so you would design the solution so that the value for which the square root is being determined is obtained via INPUT, not as a known.

Finally, we have a standard format in which your solution must be presented. For simplicity, we will begin with the **Simplified Standard Format** which can be found on Canvas in the Files section, Q&A folder.

Example problem

CS118 Problem Solving
Coffee

According to the International Coffee Organization[1], Brazil, Indonesia, and Vietnam were the recent largest producers of coffee:

Country	2010/2011 (1000's of bags)	2011/2012 (1000's of bags)	2012/2013 (1000's of bags)
Brazil	48,095	43,484	50,826
Indonesia	9,129	7,287	12,730
Vietnam	19,467	24,058	22,000

The ICO now needs you to prepare an update program for them. Your program will ask the user to provide 2013/2014 data for each of these three countries and display the results in a chart (see below) which includes the old and new data, and the percentage change between the 2012/2013 year and the data you just collected. The data should be requested in units of 1000's of bags, just like previous data. Store all numeric values in variables and use only variables (no constants) when printing numeric values.

Be sure your output has the same spacing and alignment as this sample run (including the spacing between input and output):

This is the "sample run"

```
Brazil's 2013/2014 exports (1000's of bags): 54098
Indonesia's 2013/2014 exports (1000's of bags): 11700
Vietnam's 2013/2014 exports (1000's of bags): 18032

Country          2010/2011   2011/2012   2012/2013   2013/2014    % Change
-------          ---------   ---------   ---------   ---------    --------
Brazil               48095       43484       50826       54098        6.4%
Indonesia             9129        7287       12730       11700       -8.1%
Vietnam              19467       24058       22000       18032      -18.0%
```

These are user-provided inputs.
They are not values produced by the solution.

Approach: Read the entire problem. This is a typical exam problem which shows a "sample run". This is what *might* show if the solution is implemented and executed, and the user typed in the inputs shown.

Starting situation: We have known information for three years (2010/2011, 2011/2012, 2012/2013) from three countries (Brazil, Indonesia, Vietnam).

Ending situation: Our solution will produce a table of data that includes known, input, and computed data.

Demo
I have implemented in Python the solution (see next page) for the Coffee problem and created an executable that you can run. Download the file `demo_coffee.pyc` from the Demos folder of Canvas. Save the file on your desktop.

Double-click the file on your desktop and run the program it contains. This program uses a "Command Line Interface", with which you may not be familiar. When the program requests an INPUT, type in a whole number (no commas, no decimals) and press the ENTER/RETURN key. Capture a screenshot of the resulting window and paste it into the document that you will submit. Close the window after you are done.

Constraints & Information: Must use known information, obtain new inputs, compute new values, produce output as table

Solution for the preceding problem (in Standard Problem Solving Format)

In some parts the exact order of steps is not critical – for example, it doesn't matter if you save Brazil data before the Indonesia data. But some order does matter: You cannot compute the percent change until you get the input from the user, for example. And you certainly cannot output the table until you have the percent changes.

(Note that every step uses at most one tool.)

In this portion, we make the known information available to the solution by saving it as variable data. As we will see later, this is not generally required with constant information, but it can be useful at times.

```
T01: DEFINE "B_2010" as Brazil 2010/2011 data
T02: DEFINE "B_2011" as Brazil 2011/2012 data
T03: DEFINE "B_2012" as Brazil 2012/2013 data
T04: DEFINE "I_2010" as Indonesia 2010/2011 data
T05: DEFINE "I_2011" as Indonesia 2011/2012 data
T06: DEFINE "I_2012" as Indonesia 2012/2013 data
T07: DEFINE "V_2010" as Vietnam 2010/2011 data
T08: DEFINE "V_2011" as Vietnam 2011/2012 data
T09: DEFINE "V_2012" as Vietnam 2012/2013 data
```

Next we obtain the information which is not known – the user must provide it

```
T10: INPUT from user: Brazil export bags as "B_2013"
T11: INPUT from user: Indonesia export bags as "I_2013"
T12: INPUT from user: Vietnam export bags as "V_2013"
```

Then we compute the values which depend on the user-provided information

```
T13: DEFINE: "B_pct" (as percent change from 2012) using B_2013, B_2012
T14: DEFINE: "I_pct" (as percent change from 2012) using I_2013, I_2012
T15: DEFINE: "V_pct" (as percent change from 2012) using V_2013, V_2012
```

Finally we have everything we need to produce the output

```
T16: OUTPUT to user: table using Brazil, Indonesia, and Vietnam variables
```

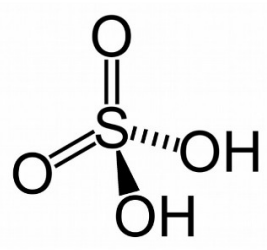
Exercise
Prepare a solution in the Simplified Standard Format for this problem:

CS118 Programming
pH

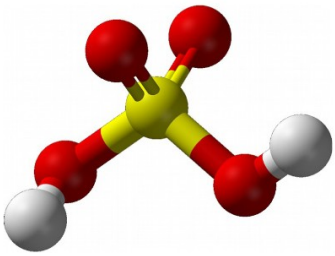
As any beginning chemistry student can tell you, the pH of a solution is a measure of the acidity of that solution. Specifically the pH is a measure of the hydrogen ion H^+ concentration:

$$pH = -\log_{10}[H^+]$$

where $[H^+]$ indicates the concentration of the hydrogen ion in moles per liter of solution.



The concentration ("molarity") of a solution is measured in moles of solute per liter of solution (M = moles/liter). A mole is a convenient abbreviation for a large quantity of atoms or compounds (specified as Avogadro's Number = 6.022×10^{23}). One mole of any compound is made up of a proportionate number of moles of its elemental components – thus one mole of H_2SO_4 (sulfuric acid) contains two moles of hydrogen atoms, one mole of sulfur atoms, and four moles of oxygen atoms. Using atomic weights, we can convert easily between moles and grams. One mole of sulfuric acid, for example, weighs about 98 grams: one mole of hydrogen weighs 1 gram, one mole of sulfur weighs 32 grams, and oxygen weighs about 16 grams per mole. (BTW, water has the molecular formula H_2O and a density of 1000g/liter – you can figure out the molecular weight yourself.)



Provide a computer program that will request the user provide grams of sulfuric acid and volume of water in milliliters, TWICE. Each should also be provided a description by the user. The program will then display the resulting molarity and pH of the sulfuric acid solution in a table as shown in the example below. Although neither is strictly correct, for simplicity you may assume the acid does not significantly impact the volume of solution and that all H^+ ions of the strong diprotic ("two hydrogen") sulfuric acid will completely separate from the compound in water solution.

```
Grams of sulfuric acid? 0.25
Milliliters of water? 5000
Description for solution #1: Dilute

Grams of sulfuric acid? 5
Milliliters of water? 100
Description for solution #2: Concentrated

Description      g H2SO4     mL H2O        g/L      M (mol/L)        pH
--------------- ----------  ----------  ----------  ----------  ----------
Dilute              0.25        5000      0.0500       0.001        2.99
Concentrated        5.00         100     50.0000       0.510       -0.01
```

Getting started – you do not need to answer any questions below; they are there to help you understand the process:

· Look at the sample run – what is the user experiencing? The first six lines of text in the sample run suggests the user is providing information; that is, he is requested to provide input. (a) What tool would our solution use to accomplish that? (b) Remember that computational tools manipulate information. We refer to that information using generic variable names that we makeup when we write our solution. What generic names will the solution use to hold the user-provided information? Those names must be part of the task that uses the tool.

· The remainder of the sample run shows a table. (c) What tool would our solution use to accomplish that? (d) Half of the information shown was obtained from the user, so the solution can only use that information by referring to the variable names created in earlier tasks.

(e) The other half of the displayed information must be computed by our solution. We can makeup names for variables that hold those values but once we've done that we will then need to make tasks to DEFINE those variables before they are used for the display. (f) What tool will define those variables in those tasks? (g) What variables will be used in those tasks? Please note: For problem solving we are NOT concerned with the exact math. We are simply saying that "computing this variable depends on this/these variable(s)". For example, we will need to compute g/L. Perhaps we decide to name that variable "`gL1`" (the "`1`" is because the problem has two of these values) While the math is simple, all we care is that to compute `gL1` we must use grams of H_2SO_4 and ml of H_2O. Assuming those values are saved in variables `gAcid1` and `mlWater1` (respectively), to compute `gL1` we would write:

```
DEFINE gL1 using gAcid1, mlWater1
```